

AD-A131 905

USER'S MANUAL FOR THE TRANSPARENT TOUCH SENSITIVE  
CATHODE RAY TUBE DISPLAY OVERLAY(U) HUMAN ENGINEERING  
LAB ABERDEEN PROVING GROUND MD M W THOMPSON JUN 83  
HEL-TN-9-83

1/1

UNCLASSIFIED

F/G 9/1

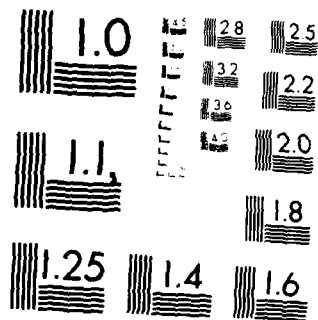
NL

END

DATE  
FILMED

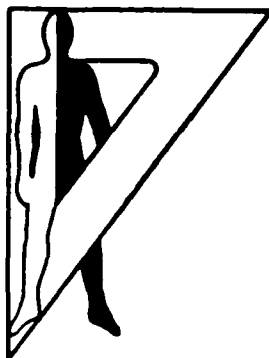
9-83

DTIC



MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

AD A131905



AD

12

Technical Note 9-83

USER'S MANUAL FOR THE TRANSPARENT TOUCH SENSITIVE  
CATHODE RAY TUBE DISPLAY OVERLAY

Michael W. Thompson

June 1983

DTIC  
ELECTE  
AUG 29 1983  
S B

Approved for public release;  
distribution unlimited.

U. S. ARMY HUMAN ENGINEERING LABORATORY  
Aberdeen Proving Ground, Maryland

83 08 19 03 6

DTIC FILE COPY

Destroy this report when no longer needed.  
Do not return it to the originator.

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Use of trade names in this report does not constitute an official endorsement or approval of the use of such commercial products.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Technical Note 9-83	2. GOVT ACCESSION NO. AD-A131905	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) USER'S MANUAL FOR THE TRANSPARENT TOUCH <del>SENSOR</del> CATHODE RAY TUBE DISPLAY OVERLAY		5. TYPE OF REPORT & PERIOD COVERED Final
7. AUTHOR(s) Michael W. Thompson		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS US ARMY HUMAN ENGINEERING LABORATORY Aberdeen Proving Ground, MD 21005		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE June 1983
		13. NUMBER OF PAGES 26
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)  Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Input Device                      XY Input Touch Panel                      Command Control Communications Touch Sensitive                      C <sup>3</sup> Interactive graphics		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This manual provides the operational and the programming information for using the Elographics Touch Sensor on the DEC PDP 11/34 computer. The Touch Sensor is a transparent pressure sensitive panel that overlays a graphics display CRT. The Touch Sensor consists of a Touch Sensor Controller and an Interface Controller for configuring the Touch Sensor to the PDP 11/34 computer. Programming is accomplished using Fortran - callable subroutines on writing MACRO language subroutines.		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

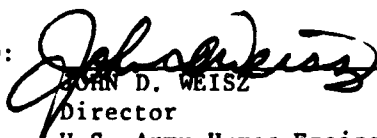
Technical Note 9-83

*TRANSPARENT*  
USER'S MANUAL FOR THE ~~TRANSPORT~~ TOUCH SENSITIVE  
CATHODE RAY TUBE DISPLAY OVERLAY

Michael W. Thompson

JUNE 1983

APPROVED:

  
JOHN D. WEISZ  
Director

U.S. Army Human Engineering Laboratory

U.S. ARMY HUMAN ENGINEERING LABORATORY  
Aberdeen Proving Ground, Maryland 21005

Approved for public release;  
distribution unlimited.

## CONTENTS

INTRODUCTION . . . . .	3
GENERAL DESCRIPTION. . . . .	3
DETAILED DESCRIPTION . . . . .	3
THEORY OF OPERATION. . . . .	6
USING THE TOUCH SENSOR . . . . .	13

## APPENDIXES

A. UNIBUS ADDRESSES AND STATUS REGISTER BIT ASSIGNMENT . . . . .	19
B. TOUCH SENSOR INTERFACE SOFTWARE USER'S GUIDE . . . . .	21

## FIGURES

1.	Position of Contact Points. . . . .	4
2.	Touch Sensor Configuration. . . . .	5
3.	Block Diagram Touch Sensor and Touch Sensor Controller. . .	7
4.	X/Y Registers & Control . . . . .	9
5.	Reg SEL FF & Command Status Register. . . . .	10
6.	M1710 Address Selection Logic . . . . .	12
7.	M1710 Unibus Interface. . . . .	14
8.	M1710 Bus Request Logic . . . . .	15

..... 14	
..... 15	✓ <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
JUL 1964	
PER CALL JC	
By _____	
Distribution/ _____	
Availability Codes _____	
Dist _____	Avail and/or Special _____
A	

## USERS MANUAL FOR THE TRANSPARENT TOUCH SENSITIVE

### CATHODE RAY TUBE DISPLAY OVERLAY

#### INTRODUCTION

The Elographics E270 Transparent Touch Sensor has been acquired for use on the Human Factors Research Simulator. An Interface Controller has been built by the US Army Human Engineering Laboratory (USAHEL) for interfacing the Touch Sensor to the Digital Equipment Corporation (DEC) PDP 11/34 minicomputer. This manual provides the basic operational and programming information for using the Touch Sensor on the DEC PDP 11/34 computer. The user should be familiar with the PDP 11/34 and the MACRO assembly language to fully understand the information in this manual. The PDP 11 Processor Handbook and the PDP 11 Bus Handbook provide a more detailed description of the PDP 11/34 computer and UNIBUS. Appendix B describes FORTRAN subroutines which provide user access to the Touch Sensor for FORTRAN program applications.

#### GENERAL DESCRIPTION

The Touch Sensor is a transparent pressure sensitive panel designed to form fit the Vector General Graphics Display System cathode ray tube (CRT). The Touch Sensor, along with the Touch Sensor Controller and the Interface Controller, provides an active area on the display surface that responds to touch when under program control. Information displayed on the CRT can be manipulated by simple finger touch action on the CRT surface. The Touch Sensor has nearly infinite resolution, but is limited to 1023x1023 area by the analog-to-digital converters of the Touch Sensor controller (Figure 1).

#### DETAILED DESCRIPTION

The Touch Sensor consists of three elements: the Touch Sensor, the Touch Sensor Controller, and the Interface Controller. All of these elements when installed and operating in the computer system require no operator action other than programming the PDP 11/34 computer for specialized use of the Touch Sensor (Figure 2).

##### Touch Sensor

The Touch Sensor is a transparent pressure sensitive panel that responds to touch when under program control. Affixed to the Vector General Display CRT, the Touch Sensor is made of a glass sheet, coated with a transparent resistive layer on one side. It is covered by a plastic

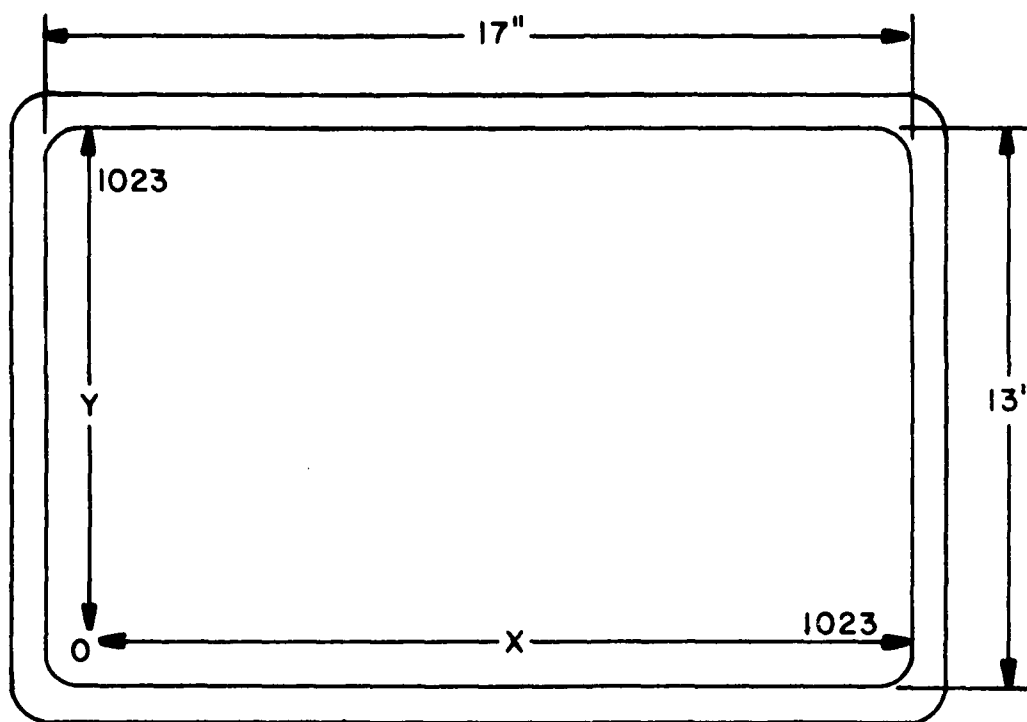


Figure 1. Position of contact points.

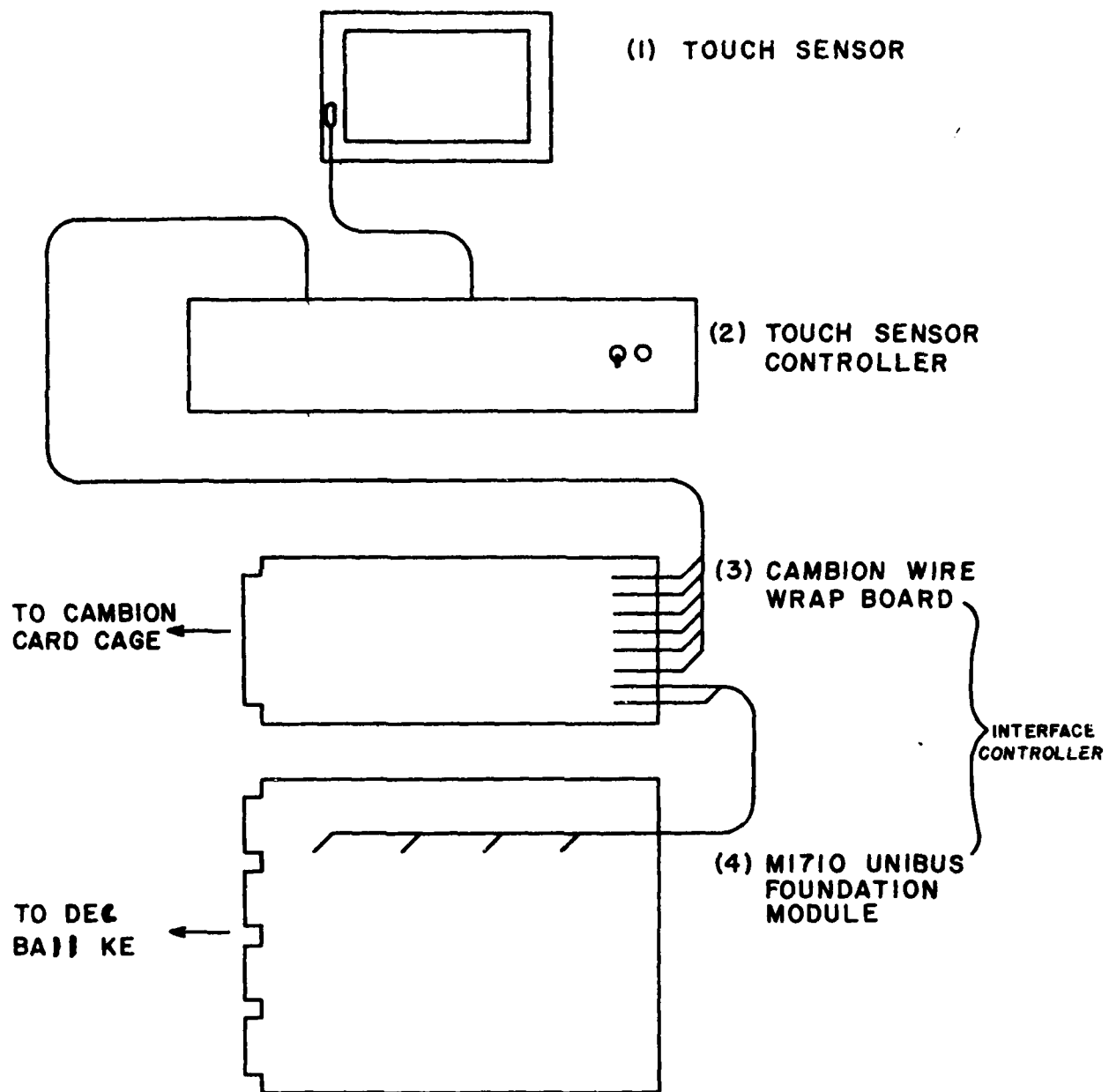


Figure 2. Touch sensor configuration.

cover sheet, coated with a transparent conductive layer on the side facing the resistive layer. Pressure applied to the plastic cover sheet causes the resistive and the conductive layers to contact resulting in a contact point being generated. Once touch down, due to finger pressure, is detected, two analog voltages are generated for the touched area representing the X and Y coordinates on the 1023x1023 resolvable area of the Touch Sensor (Figure 1).

#### Touch Sensor Controller

The Touch Sensor Controller supplies the voltages for the X and Y axis on the Touch Sensor and digitizes the analog voltage levels generated by the Touch Sensor at the point of contact. The controller provides two operational modes: single point mode that digitizes the analog voltage levels every time the Touch Sensor is touched and released, and a stream mode that digitizes at a 60-hertz rate as long as contact on the Touch Sensor is maintained. Once the analog voltages are digitized, they are ready for transfer in the form of two 10-bit parallel outputs representing the X and Y coordinates in binary form. The electronics for the Touch Sensor Controller are contained in a mounting chassis which fits in a 19" relay rack.

#### Interface Controller

The Interface Controller contains all the logic necessary for program control of the Touch Sensor on the PDP 11/34. Built around the DEC M1710 Unibus Foundation Module, the Interface Controller provides the data link between the computer and the Touch Sensor. Assigned octal device addresses from 164040 through 164044, and an interrupt octal vector address 170; the Interface Controller allows recognition of MACRO formatted instructions, thus enabling program control of the Touch Sensor by address decoding, device sensing, interrupt generation, and data transferring between the Touch Sensor and the computer.

The Interface Controller hardware consists of two circuit boards: a Cambion wire wrap board located in a 19" Cambion card cage and an M1710 Unibus Foundation Module located in the DEC Ball KE Expansion Cabinet.

### THEORY OF OPERATION

#### Analog Data Generation

A voltage is alternately applied to the X and Y axis on the resistive layer on the Touch Sensor. Finger pressure, applied at a point on the plastic cover sheet, causes the conductive layer to contact the resistive layer. This generates two voltage levels forming the X and Y coordinates in the form of analog voltage levels, one for the X axis and one for the Y axis. These analog voltage levels (analog data) are picked up by a sense line and applied to the Touch Sensor Controller (Figure 3).

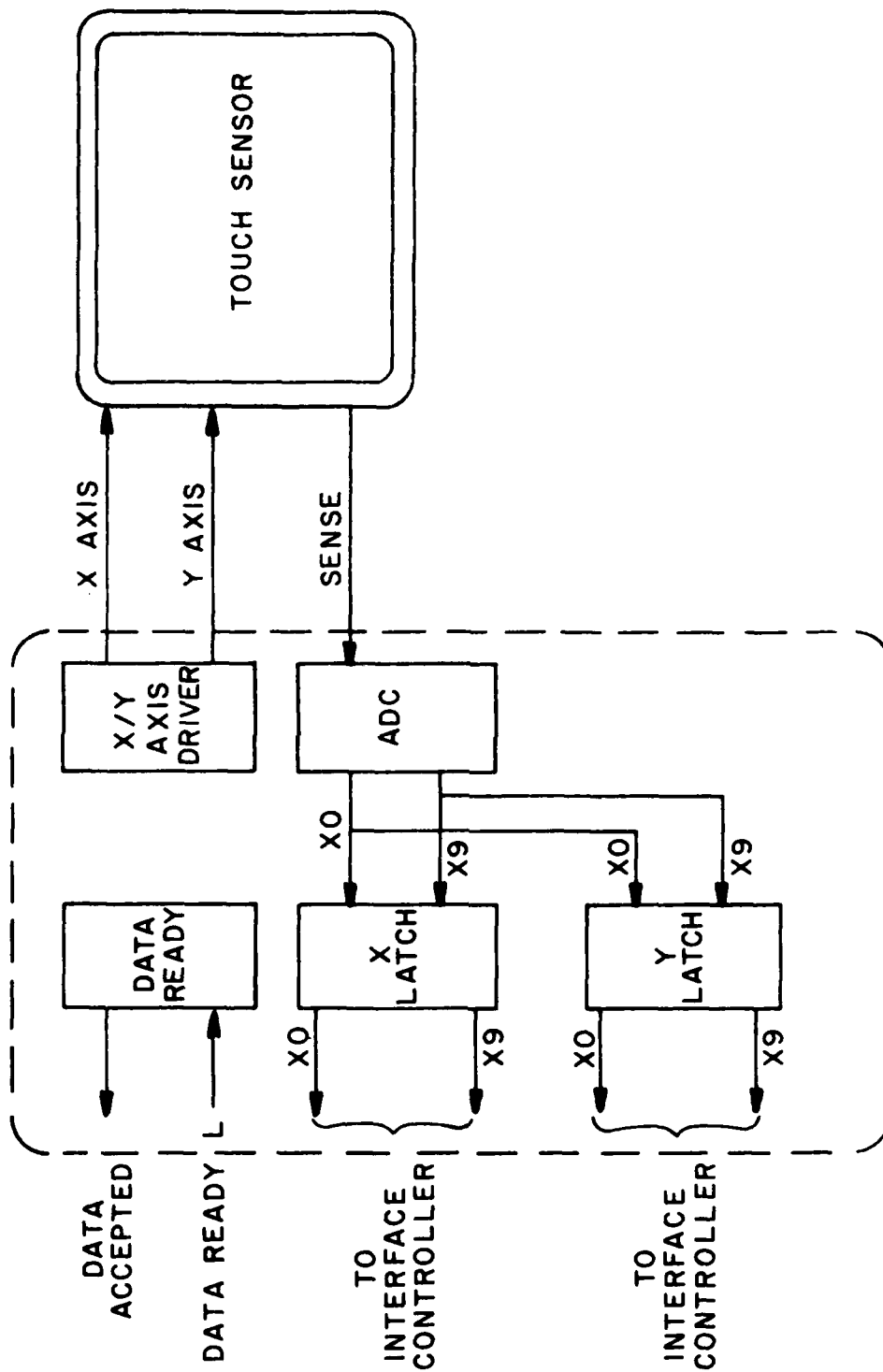


Figure 3. Block diagram Touch Sensor and Touch Sensor Controller.

The analog voltage levels are applied to the Analog to Digital Converter (ADC) located in the Touch Sensor Controller. The ADC samples the analog voltage levels, and then converts them into 10-bit binary values; one for the X coordinate address and one for the Y coordinate address. These values can range from 0 to 1023 depending upon the point of contact on the Touch Sensor. Once the X and Y values are determined, they are temporarily stored in two 10-bit registers called the X and Y Latch Registers, and a DATA READY L signal is asserted, signifying data is available to the Interface Controller (Figure 3).

#### X Register

Located in the Interface Controller on a Cambion wire wrap board, the X Register is a 10-bit temporary storage register for the binary coordinate address of the X axis prior to being placed on the Unibus of the computer. When DATA READY L is received from the Touch Sensor Controller, data from the X Latch Register is strobed to the line receivers and placed in the X Register of the Interface Controller (Figure 4).

#### Y Register

The Y Register, also located in the Interface Controller on a Cambion wire wrap board, is a 10-bit temporary storage register for the binary coordinate address of the Y axis prior to it being placed on the Unibus of the computer. When DATA READY L is received from the Touch Sensor Controller, the data from the Y Latch Register is strobed to the line receivers and placed in the Y Register of the Interface Controller (Figure 4).

Data is now in the X and Y Registers representing the point of contact on the Touch Sensor and awaiting computer action. Further action is performed according to a sequence of computer program instructions.

#### Command Status Register (CSR)

Located on a Cambion wire wrap board, the Command Status Register (CSR) is a 15-bit temporary storage register for storing the status and control information of the Touch Sensor (Figure 5). Of these, only four are of any significance to the user and are as follows: Bit 7, when set, indicates data is ready in the X and Y Registers; Bit 6, when set, enables the interrupt capability; Bit 0, when set, activates the reset command; and Bit 1, when set, activates the stream mode and when reset, activates the single point mode. Bit initiation is accomplished by executing a data transfer instruction to the CSR except for the Data Ready Bit which is initiated only by the Touch Sensor Controller. The CSR may be accessed for evaluation by executing a data transfer from the CSR to an address location selected by the user.

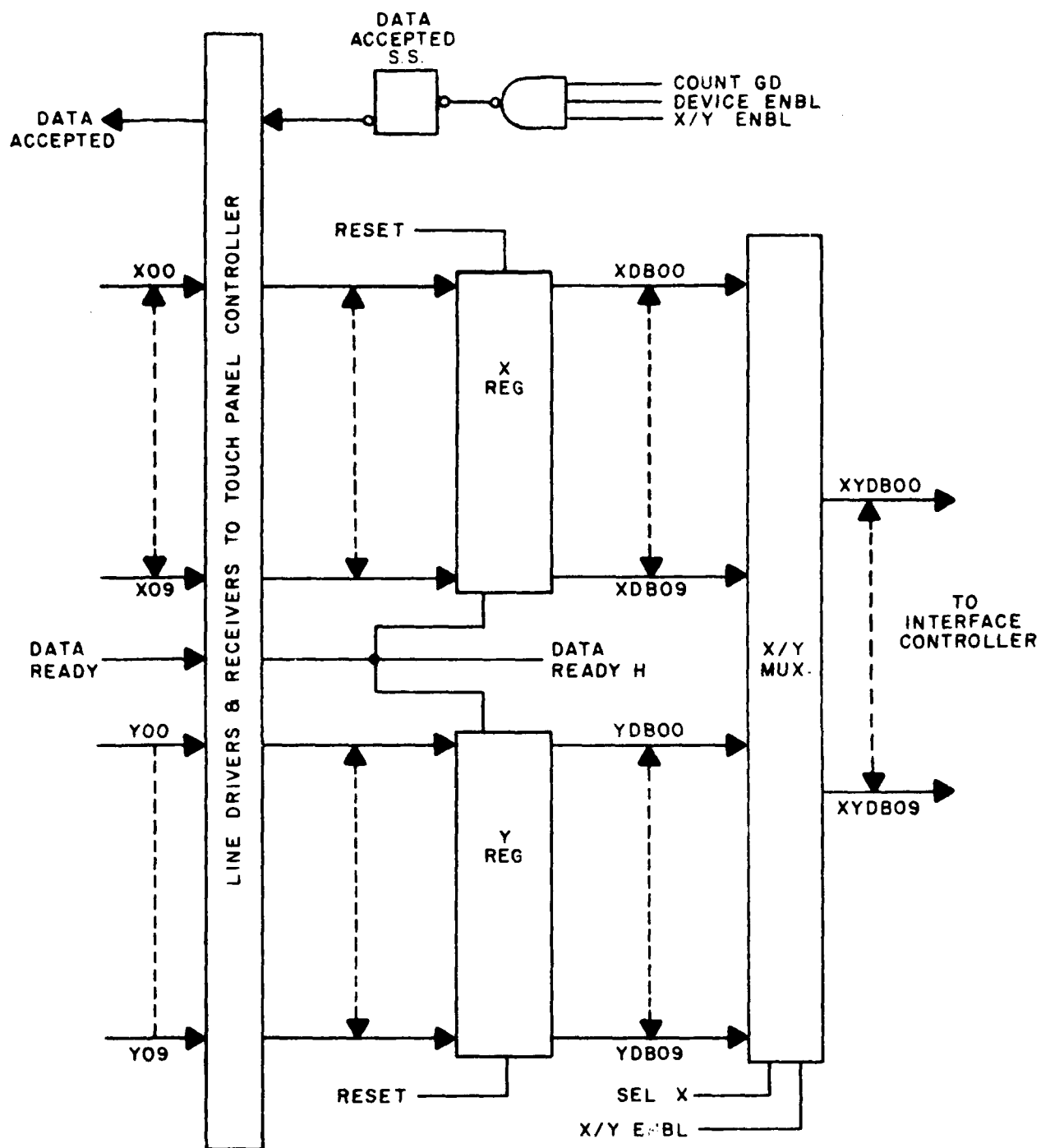


Figure 4. X/Y Registers and Control.

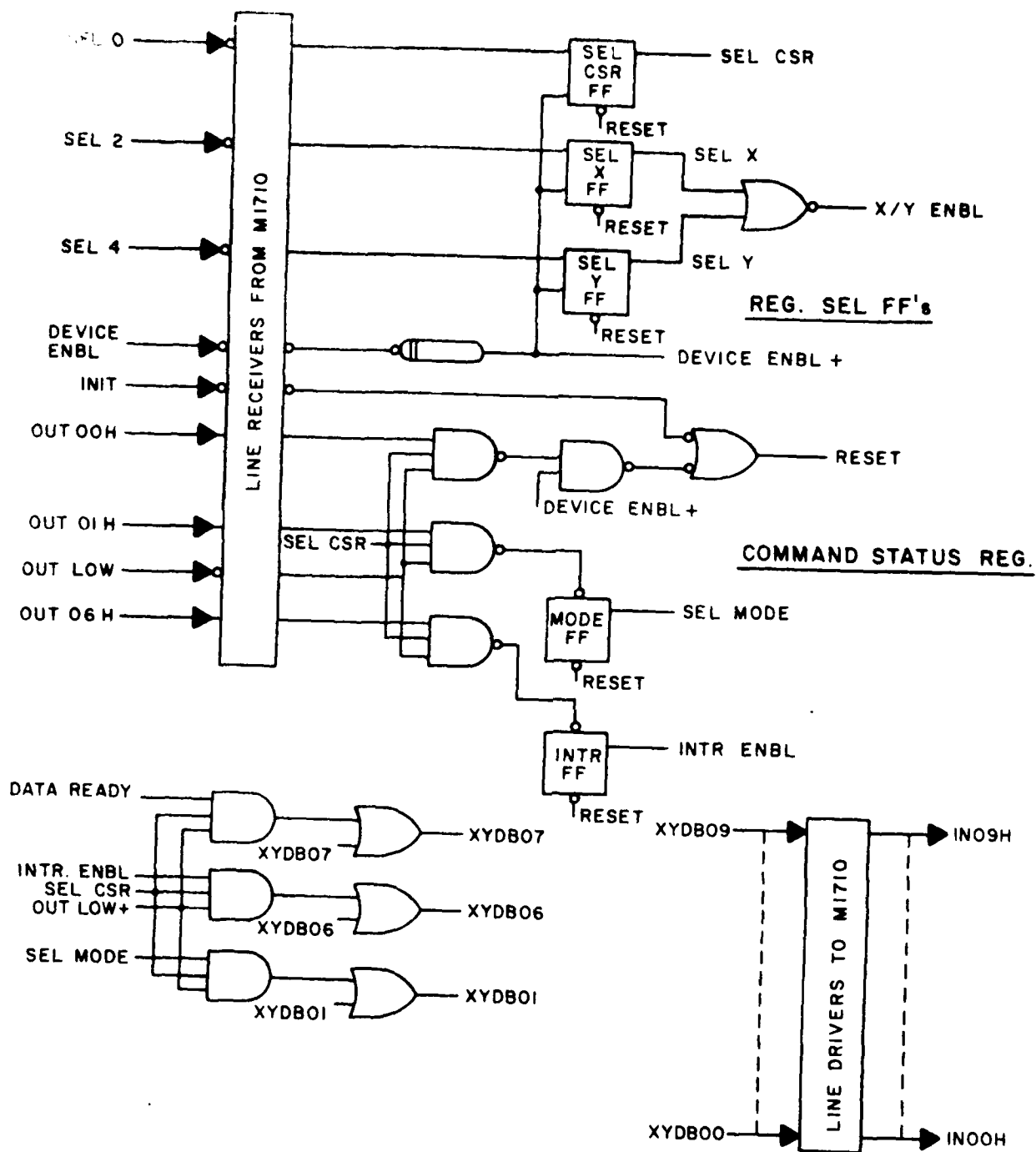


Figure 5. REG. SEL FF'S and Command Status Reg.

### Address Selection Logic

The execution of a data transfer instruction causes the appropriate address bits (A00 through A17) and the control bits (C00 and C01) to be placed on the Unibus (Figure 6). Each device on the Unibus receives the address and control bits and proceeds to decode them. After a short delay for deskew, MSYN L is asserted by the Master. Each device receives MSYN L, and after address decoding only the selected device responds, asserting DEVICE ENBL L. DEVICE ENBL L enables the 4 to 6 line decoder, and depending upon the input conditions on A01 through A04, asserts the corresponding SEL signal (Figure 6).

### Register Select Logic

When a device address is placed on the Unibus address line, address lines A1 through A4 determine the select signal (SEL) asserted which in turn selects the X Register, Y Register, or CSR Register. A binary code of 0000 placed on the address line A01 through A04 asserts SEL 0 (Figure 6). This in turn enables the SEL CSR FF to set upon the assertion of DEVICE ENBL d+, thus selecting the CSR (Figure 5).

A binary code of 0001 placed on address lines A01 through A04 asserts SEL 2. This enables the SEL X FF to be set when DEVICE ENBL d+ is asserted (Figure 6). X/Y ENBL is then asserted which places the contents of the X Register on input lines (IN00 through IN10) (Figure 5).

A binary code of 0010 placed on address lines A01 through A04 asserts SEL 4 (Figure 6). This enables the SEL Y FF to be set when DEVICE ENBL d+ is asserted. X/Y ENBL is then asserted which places the contents of the Y Register on input lines (IN00 through IN10) (Figure 5).

### Gating Control

Unibus control lines C00 and C01 designate the type of data transaction either a data-in (DATI) or a data-out (DATO) with respect to the Master device (Figure 6).

DATI transactions (C00=0 and C01=1) causes data transfers from the X Register, Y Register, or the Command Status Register to the address specified by the processor.

DATO transactions (C00=1 and C01=0) cause data transfers from the processor the Command Status Register which is the only register capable of two-way data transfers.



## Data Bus Interface

The Data Bus Interface, located on the M1710 Interface Module, contains the Unibus drivers and receivers, establishing the two-way data path between the Touch Sensor and the computer. Data to be transferred from the Touch Sensor is placed on input lines IN00 through IN10 awaiting the DRIVER ENBL L signal. Assertion of DRIVER ENBL L places the data on the Unibus for transfer to the computer (Figure 7).

## Bus Request Logic

The Bus Request Logic provides the interrupt capability for the Touch Sensors (Figure 8). INT ENBL Bit 6 of the CSR must be set to generate Bus requests. INT ENBL coupled with DATA READY H from the Touch Sensor asserts REQUEST. REQUEST asserts BR4L on the Unibus initiating the interrupt arbitration sequence. Upon completion of arbitration sequence, the requesting device (Touch Sensor) becomes master and asserts MASTER L which, in turn, asserts START INTR L and places the interrupt vector and INTRL on the Unibus data line. INTR L asserted signifies to the fielding processor that an interrupt vector is on the D lines. The fielding processor strobes the vector from the D lines and asserts SSYN L. Receipt of SSYN L causes the requesting device to remove the interrupt vector and INTR from the UNIBUS. The fielding processor then removes SSYN L informing the arbitrator that it may start issuing BGs again.

## Data Accepted

Activation of the data accepted logic requires the sequential execution of two data transfer instructions from the X and Y Registers. The first data transfer instruction partially enables activation of the DATA ACCEPTED one shot by asserting DEVICE ENBL and X/Y ENBL L. Completion of the second data transfer instruction asserts COUNT GD which enables the DATA ACCEPTED one shot. Assertion of DATA ACCEPTED then resets the DATA READY FF in the Touch Sensor Controller, making it available for processing another set of X and Y coordinate values.

## USING THE TOUCH SENSOR

Using the Touch Sensor requires writing MACRO language subroutines or using FORTRAN callable subroutines as specified in Appendix B.

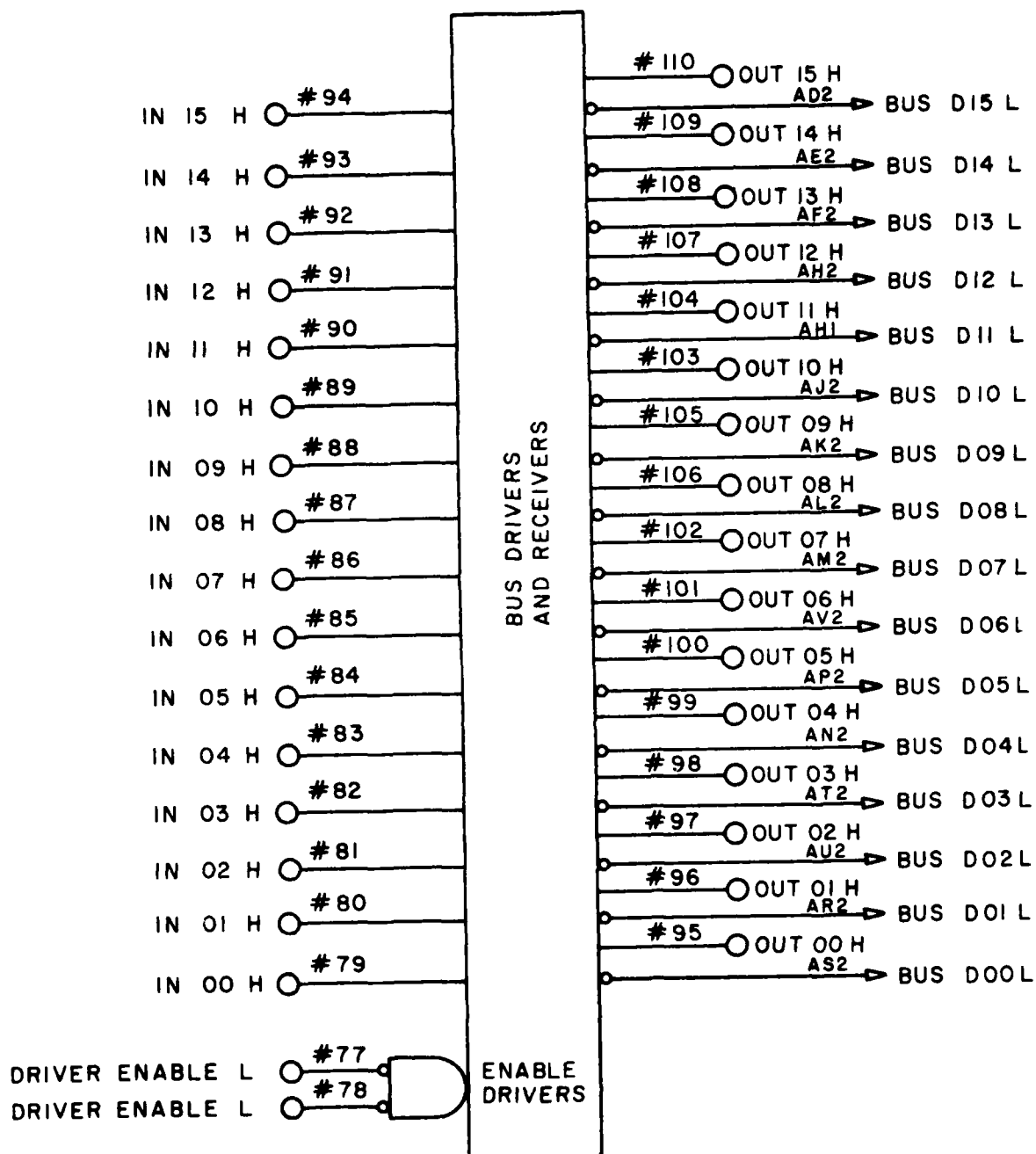
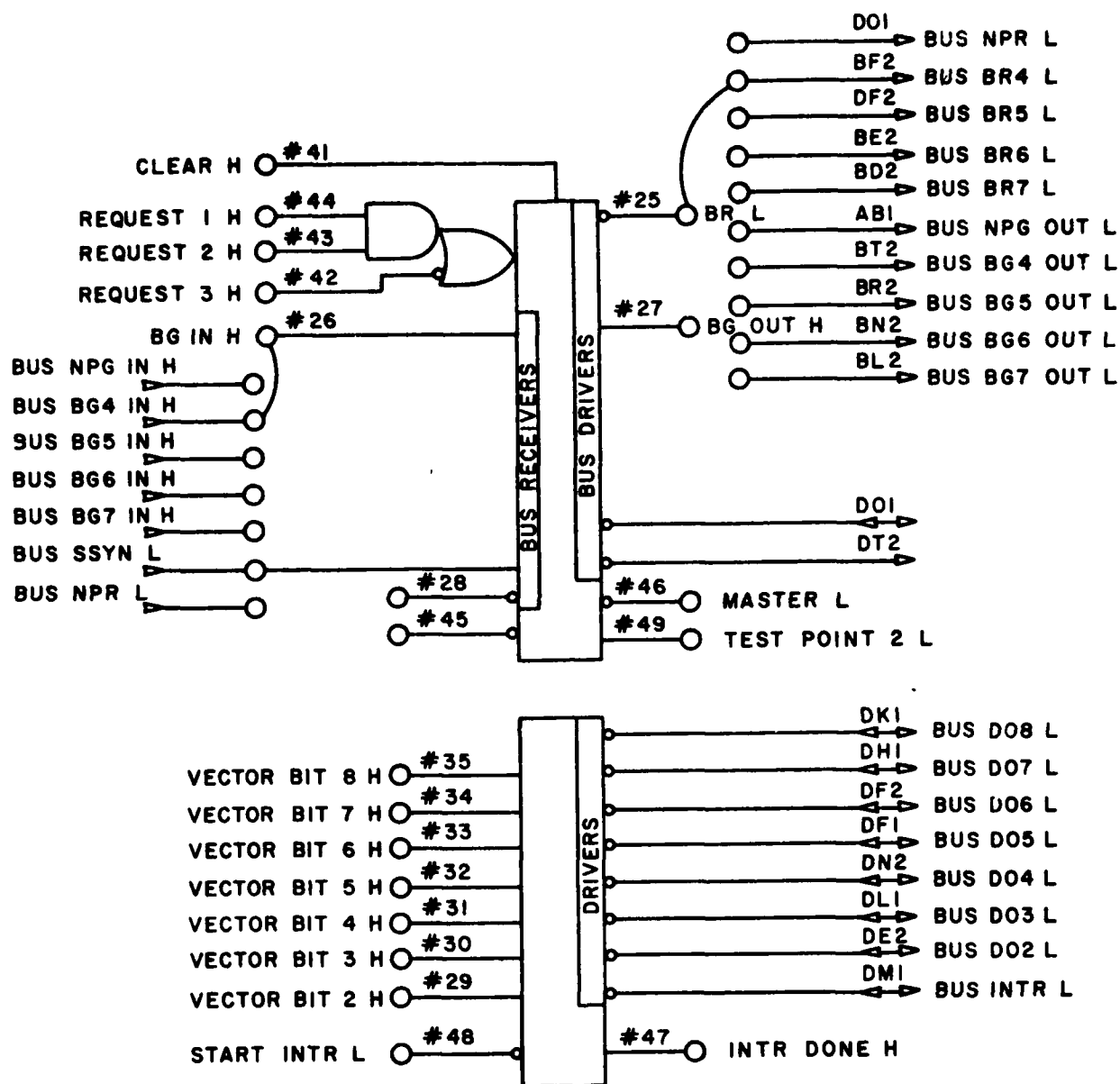


Figure 7. M1710 Bus Drivers and Bus Receivers.



M1710  
Figure 8. Bus Request Logic.

## Programming Guidance for the Touch Sensor

### Initiate the System

Before using the Touch Sensor, a programmed reset instruction should be executed which clears the X Register, the Y Register, and the CSR. It is accomplished by executing a data transfer instruction to the CSR. Once executed, repetition of this instruction is not necessary in the program sequence.

An example of executing a programmed reset is as follows:

```
MOV #1, 1764040; Set Bit 1 of CSR.
```

### Selecting the Operating Mode

The Touch Sensor has two operating modes that can be selected under program control: the single point mode, and the stream mode. If the single point mode is desired, no further action is required since a reset instruction initiates the Touch Sensor in this mode.

If the stream mode is desired, a data transfer must be executed to set Bit 2 of the CSR. To exit the stream mode, execute either the RESET instruction which clears all bits in the CSR, or a data transfer instruction which resets Bit 2 of the CSR.

The following are examples of selecting the stream mode and selecting the single point mode:

```
MOV #2,@#164040; Set Bit 2 of CSR Selecting Stream Mode
```

```
MOV #0,@#164040; Reset Bit 2 of CSR Selecting Single Point Mode
```

### Sense Data Ready

Before transferring the data in the X and Y Registers, the CSR must be interrogated by executing a data transfer instruction from the CSR 164040, and interpreting the status of the Ready Bit 7 of the CSR. Bit 7 being set indicates that the X and Y coordinate data is ready to be transferred from the Touch Sensor.

### Select X and Y Registers

After determining that data is ready (see Sense Data Ready), the X and Y Registers may now be selected. The X and Y Registers must be accessed by two data transfer instructions executed in sequence. This is required to reset the Touch Sensor for continued interaction. The following is an example of accessing the X and Y Registers:

```
MOV @#164042,R1; Transfers X Data to R1
```

```
MOV @#164044,R2; Transfers Y Data to R2
```

This sequence of instruction places the X coordinate in R1 and the Y coordinate in R2.

Once the X and Y Register contents are transferred to the computer, the Touch Sensor is ready to process another set of coordinate addresses.

#### Interrupt Request

To generate interrupts with the Touch Sensor, Bit 6 of the CSR must be set by a data transfer instruction. The following is an example of setting the interrupt enable bit:

```
MOV #100,@#164040; Set Bit 6 of CSR
```

Once set, any finger touch interaction on the Touch Panel generates interrupts using an interrupt vector of 170.

The interrupt routine should then service the X and Y register with two consecutive MOV instructions. Completion of the second MOV instruction resets the Touch Sensor for continued touch interaction.

## APPENDIX A

### UNIBUS ADDRESSES AND STATUS REGISTER BIT ASSIGNMENTS

Device Register	Address	Int. Vector	Priority
CSR	764040	Select from available floating vectors.	
X Reg	764042		
Y Reg	764044		

#### COMMAND STATUS REGISTER BIT ASSIGNMENTS:

BIT #	07	06	05	04	03	02	01	00	Function
	X	X					1		Reset
	X	1							Int. Enable
	1	X							Ready
	X	X				1			Stream

APPENDIX B

TOUCH SENSOR INTERFACE SOFTWARE  
USER'S GUIDE

Prepared By

S. Sanders  
System, Inc.  
5020 Sunnyside Ave.  
Beltsville, MD 20705  
Contract #DAAK11-83-D-0001  
Task #1

January 1983

## TABLE OF CONTENTS

INTRODUCTION. . . . .	23
REQUIREMENTS AND RESTRICTIONS . . . . .	23
INTERFACE DESCRIPTION . . . . .	24
Subroutine TSREAD. . . . .	24
Calling Sequence. . . . .	24
Usage . . . . .	25
Subroutine TSNOW . . . . .	25
Calling Sequence. . . . .	25
TASKBUILDING. . . . .	26

## TOUCH SENSOR INTERFACE SOFTWARE

### USER'S GUIDE

#### INTRODUCTION

The Touch Sensor Interface Software provides easy access to the Elo-graphics Model E-2740 Transparent Touch Sensor from FORTRAN programs. The touch sensor hardware is interfaced to the PDP-11/34 minicomputer via an interface controller which was developed by the U.S. Army Human Engineering Laboratory (USAHEL).

The Touch Sensor Interface Software consists of one relocatable object module, TSINT.OBJ, which contains two FORTRAN-callable subroutines. This document describes those subroutines and the procedures necessary to incorporate them into a FORTRAN task.

#### REQUIREMENTS AND RESTRICTIONS

The Touch Sensor Interface Software uses the Connect to Interrupt Vector (CINT\$) directive to gain access to the touch sensor hardware. Some of the code within the interface module executes as if it were part of the RSX-11M executive. This leads to the following requirements and restrictions.

For any task which uses the Touch Sensor Interface Software:

- a. The task must be built as a privileged task (/PR:0 switch required in TKB command). This implies that the task must be installed if it is to be run from a nonprivileged terminal.
- b. The task must be rebuilt after each system, using the new executive symbol table file (RSX11M.STB). Failure to do so will cause the system to crash when the task tries to access the touch sensor.
- c. The task will automatically become checkpointable after a call to the touch sensor interface, regardless of its state prior to the call. The task will be noncheckpointable while executing within the touch sensor interface.
- d. Local event flags 23 and 24 are used by the touch sensor interface routines and may not be used elsewhere in the program.
- e. Code must be executed at the priority level of the task for each touch point read in stream mode. Therefore, the priority of the task must be sufficiently high to guarantee that the maximum delay in execution of the task is less than 33 milliseconds to avoid missing data points.

## INTERFACE DESCRIPTION

The touch sensor has two modes of operation termed single-point mode and stream mode. Single-point mode is used to read the coordinates of a single touchpoint. Stream mode is used to read the coordinates of a series of points as would be generated by an operator wiping a pointer across the screen. No data is generated in either of these modes until the screen is touched. In stream mode, the touch coordinates are read at the 60 hertz rate as long as the screen is being touched. Both input modes are supported by subroutine TSREAD.

Subroutine TSNOW permits program-controlled polling of the touch sensor interface to determine if it has been touched and to obtain the coordinates of the last touchpoint.

All touchpoint coordinates are returned as a pair of 16-bit integers, each with a range of 0 to 1023. The first integer of the pair contains the x coordinate (horizontal) and the second contains the y coordinate (vertical). The lower, left-hand corner of the screen is the origin and has x and y coordinate values of zero.

### Subroutine TSREAD

TSREAD is called to read the coordinates of the next point touched, or of a series of points. TSREAD allows other tasks to execute while it is waiting for input from the touch sensor.

### Calling Sequence

CALL TSREAD (IBUF,IPTS,ITIMO,ISTAT)

IBUF	An array which will receive the x and y coordinate data: Two words in IBUF are used for each point read. The x coordinate value is stored in the first word and the y coordinate data is stored in the second word of each pair.
IPTS	Number of points to be read: Note that the size of IBUF must be at least twice the value specified for IPTS.
ITIMO	Timeout value, in seconds: Maximum time for the entire operation to complete. If ITIMO is set to zero, the operation will not be timed.
ISTAT	Completion Status: A positive value indicates successful completion and provides the number of points actually read. If a stream mode read operation times out before IPTS points are read, ISTAT will be positive and less than IPTS. A negative value indicates an error and provides the system error code from the Directive Status Word (\$D5W).

## Usage

TSREAD enables the touch sensor for operation in the single-point or stream mode, as appropriate. Operation in the single-point mode (IPTS=1) is straightforward. Operation in stream mode, however, requires additional consideration.

A timeout value should always be specified when IPTS is greater than one. If no timeout is specified, TSREAD will not return until the number of points requested has been read. The selection of a timeout interval is application dependent. However, if too long of a timeout interval is used, there may be a significant delay from the time that the operator breaks contact with the screen until the program responds. If the timeout interval is too short, the operator may not have time to respond. One possible approach is to read the first of a series of points using single point mode with no timeout, and the rest of the points using a series of stream mode reads with relatively short timeouts.

## Subroutine TSNOW

TSNOW returns the current contents of the touch sensor's x and y coordinate registers if the data in those registers has not already been read. TSNOW does not wait for the screen to be touched before it reads the touch sensor coordinates.

## Call Sequence

CALL TSNOW (IBUF, ISTAT)

IBUF     A two-word array which will receive the current contents of the x and y coordinate registers. The x coordinate value is stored in IBUF(1) and the y coordinate value is stored in IBUF(2).

ISTAT    Completion Status:  
          1 = Operation completed successfully.  
          0 = No current data in the x/y registers.  
          A negative value indicates an error and provides the system error code from the Directive Status Word (\$D5W).

## TASKBUILDING

The following must be considered when taskbuilding a task which uses the Touch Sensor Interface Software:

1. The task must be made privileged by including a /PR:0 switch on the task file specification.
2. The touch sensor interface module, TSINT.OBJ, must be included.
3. The current executive symbol table file, LB: 1,54 RSX11M.STB must be included.

The touch sensor interface module may be included in the System Library, in which case no special action is required to have it included automatically in the task.

A sample taskbuilder command line to build a task consisting of one FORTRAN module follows:

```
TKB TSTSK/PR:0=TSTSK,LB: 1,54 RSX11M.STB
```

This example assumes that TSINT.OBJ has been included in the System Library.

**DATE**  
**ILME**